



# Routage hybride sur Structure Virtuelle dans les Réseaux Mobiles Ad Hoc

Fabrice Theoleyre, Fabrice Valois

## ► To cite this version:

Fabrice Theoleyre, Fabrice Valois. Routage hybride sur Structure Virtuelle dans les Réseaux Mobiles Ad Hoc. CFIP - Colloque Francophone sur l'Ingénierie des Protocoles, Mar 2005, Bordeaux, France. inria-00406125

**HAL Id: inria-00406125**

**<https://inria.hal.science/inria-00406125>**

Submitted on 6 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Routage Hybride sur Structure Virtuelle dans les Réseaux Mobiles Ad-Hoc

**Fabrice Theoleyre — Fabrice Valois**

*Laboratoire CITI - projet INRIA ARES*

*INSA Lyon*

*21, Avenue Jean Capelle, 69621 Villeurbanne Cedex, France*

*{fabrice.theoleyre, fabrice.valois}@insa-lyon.fr*

---

*RÉSUMÉ. Le routage est le problème essentiel dans les réseaux ad-hoc. Cependant, les approches de routage à plat, tant réactives que proactives, présentent une faible capacité à passer à l'échelle : les performances décroissent rapidement dès que le nombre de participants ou de communications simultanées devient important. Par ailleurs, les mécanismes d'auto-configuration de réseaux ad hoc permettent de structurer le réseau en créant une hiérarchie. Nous proposons ici un nouveau protocole de routage, Virtual Structure Routing (VSR), tirant parti d'une auto-organisation de type topologie virtuelle constituée d'une dorsale et de zones de services. La dorsale permet de collecter le trafic de contrôle et d'en limiter le surcoût en messages. VSR utilise les zones de service pour définir une route comme une suite d'identifiants de zones. Les zones de service permettent de créer une vue logique du réseau plus stable que la topologie physique, créant des routes plus persistantes dans le temps et donc plus robustes. De plus, VSR combine les avantages des deux approches à plat : le routage intra-cluster est proactif tandis que celui inter-cluster est réactif. Enfin, nous proposons également un mécanisme de reconstruction de routes afin de limiter les pertes de paquets.*

*ABSTRACT. Routing protocols are the main issue of ad hoc networks. Because flat propositions (reactive, proactive) are not sufficient and suffer of lack of performance, new solutions should be investigated and proposed. In the other hand, self-organization propose to structure the network by creating a hierarchy between the strongest and the weakest nodes. We propose a new routing protocol, Virtual Structure Routing (VSR), based on a self-organization point of view: a virtual topology including both a backbone and clusters. The backbone is used to collect control traffic and to reduce overhead for route discovering. VSR uses clusters to define a route as a list of clusters Id. This cluster topology is more stable than the physical topology. Hence, routes are more robust. VSR combines the assets of both flat approaches: intra-cluster routing is proactive while inter-cluster routing is reactive. Finally, routes are computed dynamically and a mechanism of route repair is proposed.*

*MOTS-CLÉS : routage, réseaux ad-hoc, auto-organisation, dorsale, grappes*

*KEYWORDS: routing, ad-Hoc networks, self-organization, backbone, clusters*

---

## 1. Introduction

Les réseaux ad hoc sont littéralement des réseaux *prêts à l'emploi* : tous les terminaux peuvent communiquer les uns avec les autres par l'intermédiaire de liaisons radio, sans infrastructure fixe, et donc ne nécessitent pas le déploiement d'équipement d'interconnexion. De plus, le réseau doit pouvoir s'auto-configurer, sans aucune intervention humaine : les membres du réseau doivent collaborer pour attribuer des adresses, pour s'entendre sur les fréquences radio à utiliser. . . Un terminal souhaitant envoyer un paquet peut ne pas être à portée radio de la destination de ce paquet. Les autres membres doivent donc aider cette source à relayer les paquets vers la destination. Le réseau doit établir une coopération pour échanger les informations nécessaires à l'administration du réseau. Ainsi, les réseaux ad hoc suppriment la notion d'équipement client et d'équipement dédié : chaque terminal est à la fois client et routeur. Enfin, les terminaux sont libres de se déplacer. Le réseau subit donc continuellement des changements de topologie auxquels il doit s'adapter en mettant à jour ses informations. Les réseaux ad hoc de par leur flexibilité, sont promis à un spectre d'utilisation très large. Ils peuvent être utilisés pour des applications militaires sur des champs de bataille, permettant d'acheminer les informations entre fantassins, véhicules et centre de commandement, sans liaison satellite systématique à travers le déploiement rapide d'infrastructure de communication. De façon plus large, nous pensons également que les réseaux ad hoc permettront d'étendre la notion de réseau sans fil cellulaire, en proposant une couverture sans fil multisauts, étendant la couverture radio à un moindre coût, créant de véritables *réseaux d'accès sans-fil multisauts*.

Les réseaux ad hoc, de par leur manque d'infrastructure dédiée et leur dynamique, demandent de repenser totalement les solutions existantes dans les réseaux filaires. L'utilisation d'un médium radio présente de fortes contraintes en terme de bande passante disponible, d'interférences engendrées, d'instabilité du médium, de fiabilité et de pertes de paquets. . . Parallèlement, la mobilité rend la topologie fortement dynamique, obligeant le réseau à se reconfigurer logiquement pour s'adapter. Une telle reconfiguration doit être la plus rapide possible, tout en limitant au minimum les échanges de paquets de contrôle. Enfin, l'inondation est très utilisée dans les réseaux ad hoc pour diffuser une information dans le réseau. Cependant, les inondations créent dans les réseaux ad hoc un problème de redondance de transmissions et de manque de fiabilité, connu sous le problème de la tempête d'inondation [NI 99]. Il existe dans les réseaux ad hoc de nombreuses problématiques à aborder : affectation d'adresses, interconnexion à un réseau filaire de type Internet, sécurisation des échanges, qualité de service. . . Cependant, le routage constitue le problème fondamental : il est nécessaire de proposer une solution permettant d'acheminer un paquet envoyé par une source, jusqu'à une destination. Ce routage doit être réalisé en minimisant le délai et les pertes, et ce, en présentant un échange minimal en paquet de contrôle. De plus, un protocole de routage doit pouvoir s'adapter aux changements de topologie rapidement, en proposant des routes de longueur acceptable, même en cas de forte mobilité des terminaux.

Notre point de vue est qu'il faut d'abord structurer, organiser le réseau avant de pouvoir diffuser efficacement une information, de router des paquets et de proposer

des services supplémentaires (i.e. localisation). L'organisation permet de tirer partie de l'hétérogénéité des nœuds en termes de puissance, énergie, etc. La vision de l'auto-organisation que nous défendons se base sur la notion de topologie virtuelle proposée dans [THE 04b]. Nous appelons topologie virtuelle une organisation basée sur une épine dorsale maintenue dynamiquement sur laquelle vient se greffer des grappes qu'elles soient géographiques ou liées à des services. Cette topologie offre alors une vue simplifiée et plus exploitable du réseau. La dorsale permet l'optimisation de la diffusion d'information dans le réseau, en n'impactant que les membres de la dorsale. Les grappes permettent de créer une vue logique plus stable de la topologie, en groupant les nœuds dans une zone selon des contraintes de persistance. Un nœud peut ainsi physiquement se déplacer, mais rester dans la même zone de service, et donc ne crée pas de changement dans la topologie logique. De plus, un chef dans chaque zone permet de coordonner les fonctions de routage ou d'affectation d'adresses. Les nœuds membres de la dorsale ou chefs de zones sont choisis pour leur stabilité et leur aptitude de coordinateur, transparaissant à travers notre métrique de stabilité. De telles topologies sont paramétrables : le nombre de membres de la dorsale et le nombre de zones de services sont modifiables pour adapter la topologie à une application ou à un environnement différents. Cependant, les réseaux ad hoc sont des environnements volatiles. Nous avons donc également proposé une procédure de maintenance pour garder une structure efficace tout au long de la durée de vie du réseau. [THE 04a] propose des améliorations en optimisant la robustesse de nos structures, tout en réduisant l'overhead. Nous avons également présenté une solution permettant de redonder les points d'accès, passerelles vers Internet.

Nous proposons ici un nouveau protocole de routage basé sur une telle solution d'auto-organisation combinant les avantages des protocoles de routage proactifs et réactifs. Le routage à l'intérieur d'une grappe est proactif : chacun connaît une route vers chacun des membres de sa grappe. Parallèlement, le routage inter-grappes est réactif : un nœud va utiliser la dorsale pour minimiser le volume de trafic engendré par une demande de route. De plus, la route s'appuie sur la vue en grappes du réseau, la route étant constituée d'une suite de grappes à suivre. La route est donc plus robuste qu'une route basée sur la topologie physique, plus volatile. Le prochain saut au niveau physique est calculé dynamiquement, afin de s'adapter au plus vite aux changements locaux de la topologie. Enfin, nous proposons également un mécanisme de reconstruction locale de route en cas de défaillance d'un nœud intermédiaire.

Nous allons dans une première partie présenter un état de l'art des solutions de routage dans les réseaux ad hoc. Ensuite, nous présenterons dans la partie 3 notre solution de routage sur structure virtuelle, en décrivant très brièvement la topologie virtuelle sur laquelle nous nous appuyons. Nous évaluerons dans la partie 4 sous forme de simulations l'efficacité de notre solution. Enfin, nous concluons dans la partie 5 en donnant quelques perspectives.

## 2. État de l'art

Essentiellement deux approches traitent du routage au cœur des réseaux. L'approche à état de liens (OSPF [MOY 98]) maintient la topologie globale du réseau sur laquelle l'algorithme de Dijkstra est exécuté pour calculer des routes optimales vers chacun des membres du réseau. Chaque nœud diffuse périodiquement un paquet de présence, engendrant un coût important en messages de contrôle. A l'opposé, les approches à vecteurs de distance (RIP [HED 88]) calculent la route de proche en proche. Chaque routeur diffuse périodiquement toutes les routes qu'il connaît et leur distance. Si un routeur reçoit une route plus courte que celle qu'il connaît, il la met à jour et la diffusera dans sa prochaine annonce de routes. Cependant, des problèmes de convergence des tables de routage peuvent se poser lorsqu'un lien est supprimé. Ces approches sont inacceptables pour les réseaux ad hoc, à cause de leur lenteur de convergence ou de l'overhead excessif dû aux changements importants de topologie.

### 2.1. Routage à plat

De nombreux protocoles de routage ont été proposés. Dans l'approche *proactive*, chaque nœud connaît une route vers chaque nœud du réseau. Si chaque nœud envoie un paquet de topologie dans tout le réseau de façon périodique, l'overhead induit est élevé et une tempête d'inondation risque d'apparaître. DSDV [PER 94] est donc basé sur l'algorithme de Bellman-Ford et n'échange des paquets de topologie qu'avec ses voisins radio. OLSR [CLA 03] optimise la diffusion des paquets de topologie via un sous-ensemble de nœuds chargés de relayer seuls les inondations. Les protocoles proactifs optimisent les délais de livraison mais engendrent un overhead important.

A l'opposé, le *réactif* crée des routes *à la demande*. Un nœud initie une découverte de route lorsqu'il doit envoyer un paquet et qu'il ne connaît aucune route vers la destination. DSR [JOH 03] envoie un *Route Request* inondé dans le réseau, accumulant dans le paquet les adresses des nœuds relais. Lorsque la destination reçoit la *Route Request*, elle renvoie un *Route Reply* sur la route inverse contenue dans le paquet. La source peut donc mettre en cache la nouvelle route et commencer à envoyer ses paquets de données. Lorsque cette route n'est plus valide, une nouvelle découverte est initiée. Une telle approche permet d'optimiser la taille mémoire allouée à la table de routage, et présente un overhead réduit lorsqu'un nœud ne possède qu'un nombre réduit de correspondant, changeant peu au cours du temps. Par contre, les délais d'établissement de route peuvent être longs, ce qui peut être problématique pour des applications multimédia. AODV [PER 03] propose une approche similaire, mais en maintenant des tables de routage distribuées, supprimant le routage par la source.

### 2.2. Routage hybride

ZRP [HAA 98] propose une approche hybride combinant proactif et réactif. Chaque nœud maintient proactivement une route vers chacun des nœuds à moins de  $p$

sauts de lui,  $p$  étant le rayon de la zone. Le routage inter-zone est réactif : la source envoie une demande de route avec un paquet `Route Request` aux nœuds bordure de sa zone, i.e. à  $p$  sauts exactement. Ces nœuds bordure relaient cette requête à leurs propres nœuds bordure, jusqu'à ce que le paquet atteigne la destination. La route finale ainsi formée est constituée par les identifiants de nœuds bordure traversés, le routage entre ces nœuds étant assuré par le protocole proactif intra-zone. Cependant, les zones sont recouvrantes, puisqu'une zone n'a de signification que pour un seul nœud, les routes sont donc orientées source. La diffusion peut être pire que l'inondation pure si un nœud doit relayer un `Route Request` plusieurs fois pour des nœuds bordure différents et déjà couverts. Des techniques sont proposées dans [PEA 99] pour limiter le nombre de paquets engendrés par cette diffusion. Lorsque la densité augmente, le nombre de nœuds bordure augmente de façon drastique, impactant l'overhead. [PEA 99] propose de sélectionner les nœuds bordures adéquats, mais selon la connaissance du  $(2 \cdot p + 1)$ -voisinage, ce qui a un impact important sur l'overhead. De plus, les zones n'étant pas définies de façon hiérarchique, ZRP ne reflète pas l'hétérogénéité naturelle d'un réseau ad hoc. Une structure plus organisée serait donc peut être plus efficace.

### 2.3. Routage hiérarchique

Hiérarchiser le réseau pour le structurer est une approche intéressante pour simplifier le fonctionnement d'un réseau ad hoc. Des approches basées sur le découpage du réseau en grappes ont déjà été proposées. Dans les solutions créant des grappes avec chef, chaque nœud doit être voisin de son chef de grappe. Les algorithmes de construction de grappes sont souvent basés sur un processus d'élection des chefs de grappe, leurs voisins devenant automatiquement les membres de sa grappe. Lors de la maintenance, on peut forcer la grappe à maintenir comme chef le nœud le plus fort localement, ce qui engendre un grand nombre de fusions et fissions de grappes. Il est possible de minimiser les changements : lorsqu'un nœud se retrouve isolé, un nouveau chef est élu, et seulement dans ce cas là. Dans les grappes sans chef explicite, chaque nœud doit être au maximum à 2 sauts de tous les autres membres de sa grappe, i.e. le diamètre de la grappe doit être de 2 sauts. La maintenance est dans ce cas là plus complexe. Lorsque la contrainte de diamètre est violée, il est nécessaire de décider de façon distribuée de la méthode de fission de la grappe pour minimiser le nombre de créations de nouvelles grappes ou le nombre de nœuds changeant de grappe.

CBRP [JIA 99] propose une solution de routage sur grappes. Des chefs de grappes sont élus et sont chargés de coordonner le routage dans leur grappe. Chaque nœud envoie périodiquement des paquets `hello`s contenant la liste de ses 1-voisins, et la liste des grappes adjacentes. Ainsi, un nœud a la connaissance de son 2-voisinage, et les chefs de grappes connaissent en plus l'identité des grappes adjacentes et l'identifiant d'un nœud passerelle pour les atteindre. L'ensemble des nœuds passerelle et des chefs de grappe forme un sous-ensemble connexe du réseau.

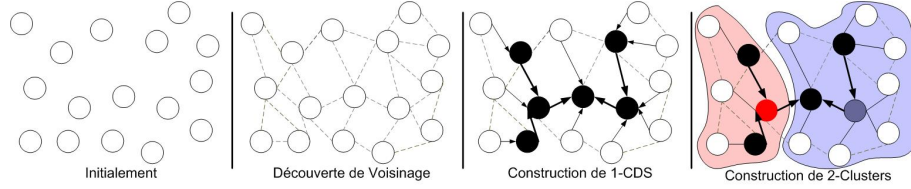
Lorsqu'un nœud souhaite envoyer un paquet de données et qu'il ne connaît aucune route, son chef de grappe sert de *proxy* pour la découverte de route. Le chef envoie en unicast un *Route Request* à chacune des grappes adjacentes, en choisissant pour chacune une passerelle par laquelle passer. Chaque chef de grappe recevant un *Route Request* relaie le paquet vers ses grappes adjacentes. La destination renvoie un *Route Reply* suivant la route inverse contenue dans le paquet. Chaque chef de grappe tente de calculer une optimisation de routes grâce à sa connaissance locale avant de relayer ce paquet vers la source. Cependant, la route est donnée comme une liste de nœuds intermédiaires. Il suffit donc qu'un seul lien de la route change pour que la route soit cassée. CBRP propose une solution de reconstruction locale de route : si un nœud s'aperçoit que le lien que doit emprunter le paquet est cassé, il cherche un autre nœud existant capable de relayer le paquet vers le prochain saut. La route est donc allongée d'un saut à chaque reconstruction. Enfin, l'inondation des *Route Requests* est sous-optimale car la topologie formée par l'ensemble des passerelles et des chefs de grappes est redondante, de nombreuses boucles existent, augmentant l'overhead. La création d'une structure plus évoluée pour les inondations pourrait donc être plus performante.

#### 2.4. Routage sur dorsale

Des solutions de création de dorsales ont déjà été proposées. Dans [SIV 98], les auteurs créent une dorsale connexe, telle que chaque nœud du réseau est voisin d'au moins un membre de la dorsale. Cette dorsale est chargée de collecter tous les paquets de topologie envoyés par les nœuds, et de calculer des routes pour des nœuds tiers. CEDAR [SIN 99] propose de créer une dorsale constituée de chefs de zone, interconnectés entre eux par des liens virtuels. Les chefs de zone envoient périodiquement des *hello*s à 3 sauts pour maintenir ces liens virtuels. L'overhead peut donc être important. De plus, l'environnement étant dynamique, les liens virtuels deviennent rapidement sous-optimaux. Chaque nœud est responsable de la publication de tous ses liens. Il notifie le nœud de la dorsale duquel il dépend des changements de bande passante de ce lien. Un mécanisme de vagues de diffusion sur la dorsale, avec des files d'attente différenciées permettent de ne propager loin que les liens les plus stables. Ensuite, la dorsale est chargée de calculer les routes selon la qualité de service demandée par la source. Pour ne pas engorger la dorsale, la structure n'est utilisée que pour l'établissement des routes de secours, les routes usuelles passant par des liens *normaux*.

### 3. Proposition

Notre proposition se focalise sur le problème de routage dans les réseaux ad hoc. Il nous paraît important de structurer le réseau avant de router. Nous pensons qu'il faut tirer partie de l'hétérogénéité naturelle du réseau pour hiérarchiser les nœuds, élire des coordinateurs possédant un rôle prépondérant dans le fonctionnement du réseau. Les réseaux ad hoc souffrent principalement de la faiblesse de bande passante et des



**Figure 1.** Schéma de la construction de la structure virtuelle

changements de la topologie due à la mobilité des terminaux. Nous proposons donc d'organiser le réseau pour répondre à de telles problématiques.

Dans un premier temps, nous souhaitons optimiser la diffusion d'informations dans le réseau en limitant le nombre de nœuds chargés de stocker l'information et de relayer des paquets. Dans ce but, nous utilisons une dorsale virtuelle, formant une véritable prolongation des dorsales présentes dans les réseaux filaires. Ce sous-ensemble de nœuds sera chargé du relais de paquets de contrôle diffusés. Parallèlement, nous souhaitons fractionner notre réseau en entités logiques plus stables que la topologie physique, à la manière des sous-réseaux du domaine filaire. Dans ce but, nous élisons donc des chefs de grappes, créant des zones logiques pouvant se transformer en véritables zones de service. Un nœud peut physiquement changer de voisins, mais ne pas changer de grappes. La vue logique est donc plus stable que la topologie physique. Les routes seront donc définies sur une telle structure pour tirer partie de sa robustesse. Nous allons tout d'abord introduire brièvement la structure virtuelle utilisée. Puis nous décrirons notre solution de routage.

### 3.1. Topologie Virtuelle

Nous utilisons la structure décrite dans [THE 04b]. Dans un premier temps, chaque nœud initie une découverte du  $k_{cds}$ -voisinage (les nœuds à moins de  $k_{cds}$  sauts radio) en envoyant la liste de ses voisins dans un hello avec un *ttl* fixé à  $k_{cds}-1$ . Un nœud peut donc connaître l'identité de ses  $k_{cds}$ -voisins, l'adresse d'un nœud intermédiaire pour les joindre, et peut différencier les liens bidirectionnels des liens unidirectionnels.

Ensuite, nous construisons une dorsale en sélectionnant les nœuds les plus aptes à un tel rôle à travers un poids et une relation d'ordre. Nous avons proposé un poids basé sur la notion de persistance. Le leader initie la construction en devenant le premier membre de la dorsale (un dominant). Si aucun leader naturel n'existe, on exécute une élection distribuée. Grâce à la connaissance du  $k_{cds}$ -voisinage, certains des nœuds sont élus dominants de façon distribuée pour former un  $k_{cds}$ -CDS (Connected Dominating Set), i.e. une structure connexe de *dominants* telle que tout nœud est au maximum à  $k_{cds}$  sauts d'au moins un dominant. Les nœuds non membres de la dorsale sont appelés *dominés*. La dorsale forme ainsi un arbre de dominants dont les feuilles sont les dominés, et la racine est le leader. Chaque nœud enregistre l'identité de son père



dans l'arbre, et chaque dominant enregistre celle de ses fils, i.e. les dominants desquels il est père. Sur cette dorsale, nous découpons le réseau en grappe. Certains dominants sont élus chefs de grappes en fonction de leur poids, leurs voisins dominants se ralliant aux nouvelles grappes ainsi créées. Seuls les dominants participent à un tel processus, réduisant l'overhead, et forçant un chef de grappe à être membre de la dorsale pour optimiser plus tard la diffusion d'informations.

Les réseaux ad hoc constituent des environnements volatiles. Ainsi, nous avons proposé un protocole de maintenance. Une procédure classique de découverte de voisinage utilisant des `hello`s sert à maintenir l'adresse, l'état et le poids de chacun des  $k_{cds}$ -voisins. Les dominés peuvent ainsi trouver de façon distribuée un nouveau dominant pour se reconnecter à la dorsale. De même, les dominants peuvent retrouver un nouveau père pour se reconnecter à la dorsale. Ainsi, le leader envoie périodiquement des `ap-hello`s, relayés seulement par les dominants. Ces paquets permettent aux dominants de retrouver localement un point de reconnexion dans la dorsale lorsque la topologie a changé. Dans chaque `hello`, un nœud place l'identité de son chef de grappe et sa distance. Ainsi, tout dominant peut vérifier périodiquement la validité de son chef de grappe et retrouver un nouveau chef si besoin est. Une telle maintenance est rendue possible par la forme en arbre de la dorsale, et donc l'absence de boucle, supprimant le problème de l'horizon infini de Bellman-Ford [HED 88]. Nous avons montré qu'une telle topologie virtuelle ainsi construite présente une très grande stabilité : à  $30\text{m.s}^{-1}$ , plus de 99% de taux de connexion pour la dorsale et les grappes et un chef de grappe reste chef pendant plus de 2 minutes [THE 04a].

### 3.2. Routage Intra-Grappe

Nous supposons que les nœuds vont, dans les réseaux ad hoc ou hybrides, principalement communiquer avec des nœuds proches géographiquement, et quelquefois avec des nœuds distants. Par exemple, une personne sur un campus collaborera activement avec ses collègues, et accédera sporadiquement à des ressources distantes présentes par exemple sur des serveurs de fichiers. Ainsi, il serait intéressant de maintenir une connaissance locale proactive de la topologie. Nous avons donc choisi de proposer un protocole de routage proactif au sein d'une grappe. De plus, une telle solution nous permet d'exécuter un autre protocole de routage sur la topologie en grappes. Cette topologie étant plus stable que le voisinage physique, la solution inter-grappe est donc plus robuste, et demande un overhead plus réduit. En paramétrant la valeur du rayon de nos grappes, nous pouvons ainsi réguler l'overhead induit par une solution proactive. Un nœud doit donc connaître une route vers chacun des membres de sa grappe.

Le protocole de maintenance de la topologie virtuelle crée déjà une table de voisinage du  $k_{cds}$  voisinage. Un rayon de dorsale  $k_{cds}=2$  représente une dorsale stable avec un nombre réduit de chefs de grappe ( $\approx 20\%$  [THE 04a]). De plus, pour détecter les liens unidirectionnels, un nœud est obligé de placer ses 1-voisins dans les `hello`s. La transmission d'un `hello` en broadcast par chaque nœud est donc suffisant pour connaître le 2-voisinage. Nous proposons donc de fixer le TTL à  $2 \cdot k_{cluster} + 1$ . Un

nœud relaie un `hello` s'il vient d'un lien bidirectionnel, d'un nœud appartenant à la même grappe, et que le TTL est supérieur à 1. Cette découverte de voisinage permet bien de connaître la topologie de la grappe. Sur cette topologie, un nœud peut exécuter l'algorithme de Dijkstra pour calculer les routes optimales. Il peut également enregistrer l'adresse du nœud lui ayant fourni le `hello` de  $S$  comme prochain saut pour atteindre  $S$ . Les tables de voisinages contiennent des informations sur des nœuds proches géographiquement. Elles convergent donc rapidement, et sont peu impactées par la mobilité. Si on suppose qu'elles sont cohérentes au sein d'une zone, chaque nœud peut calculer de façon cohérente une route, saut par saut.

### 3.3. Routage Inter-Grappe

Nous proposons d'exécuter un protocole de routage réactif entre les grappes. Un nœud découvre une route à la demande lorsqu'il doit contacter un nœud pour lequel il ne connaît aucune route. La route est définie sur la topologie en grappes, plus stable. La découverte de routes utilise la dorsale pour optimiser la diffusion de la requête.

#### 3.3.1. Découverte de la topologie en grappes

La route étant définie sous la forme d'une suite d'identifiants de grappes à suivre, un nœud doit connaître les grappes adjacentes, et une route pour les atteindre. Nous intégrons cette connaissance aux paquets `hello`s. Un nœud publie déjà dans les `hello`s son identifiant, son poids, son état et l'identifiant de son chef de grappe. Il va donc annoncer en plus les identifiants des grappes voisines. Un tel nœud peut donc servir de passerelle vers les grappes annoncées. Une route vers une passerelle peut d'autre part être calculée grâce au protocole proactif de routage intra-grappe. Cet accroissement en taille nous semble acceptable : en radio, le nombre de paquets a un impact plus important sur les performances que la taille des paquets.

#### 3.3.2. Découverte de routes

Lorsqu'un nœud  $S$  souhaite envoyer un paquet de données à  $D$ , les cas suivants peuvent se produire :

- $D$  est au maximum à  $k_{cds}$  sauts de  $S$  ou  $S$  et  $D$  ont le même chef de grappe.  $D$  est donc dans la table de voisinage de  $S$ .  $S$  exécute directement l'algorithme de routage proactif intra-grappe introduit précédemment pour atteindre  $D$
- $D$  est dans la table de routage de  $S$ .  $S$  a donc une route de grappes pour atteindre  $D$ . Il exécute l'algorithme de routage inter-grappe ;
- Sinon,  $S$  initie une découverte de routes.

$S$  laisse son dominant jouer le rôle de proxy pour une découverte de route. Il génère un `Route Request` dans lequel il inscrit l'adresse de son chef de grappe dans la liste des chefs de grappe du paquet. Puis il l'envoie aux autres membres de la dorsale en multicast. Les dominés ne participent donc pas à la découverte de routes. Chaque dominant recevant un `Route Request` relaie le paquet s'il ne connaît pas  $D$  et s'il n'a

jamais vu le paquet auparavant. Avant de le relayer, le dominant inscrit son identifiant de grappe s'il est différent du précédent identifiant de grappe de la route contenue dans le paquet. Si  $D$  est dans la table de voisinage du dominant, alors celui-ci génère un *Route Reply* contenant la route des grappes contenue dans le *Route Request*. De même, il ajoute dans cette route l'adresse de son identifiant de grappe et l'identifiant de grappe de  $D$  s'ils n'y sont pas déjà présents. La *Route Reply* contient donc finalement la route de grappes à suivre de  $D$  à  $S$ . Le dominant envoie la *Route Reply* vers  $S$  en exécutant l'algorithme de routage inter-grappe. Les *Route Request* ne sont relayées que par les membres de la dorsale, diminuant de beaucoup l'overhead induit par une découverte de routes.

### 3.3.3. Routage

Nous exécutons le protocole de routage inter-grappes pour les paquets de données et les *Route Reply*. La route de grappes à suivre est contenue dans l'en-tête du paquet. Avant de relayer un *Route-Reply*, un nœud peut mettre en cache la route apprise vers  $S$  et vers  $D$ , afin de réduire le nombre de découvertes de routes générées plus tard. Si la destination finale est dans la table de voisinage, alors le nœud  $N_1$  envoie directement le paquet grâce au routage intra-grappe. Sinon, il cherche la première grappe connue  $G$ , la plus proche de la destination :

- Un 1-voisin  $N_2$  possède le chef de grappe  $G$ .  $N_2$  est le prochain saut ;
- Un 1-voisin  $N_2$  est passerelle pour  $G$ .  $N_2$  est le prochain saut ;
- Un nœud  $N_2$  est la plus proche passerelle vers  $G$  de la grappe de  $N_1$ .  $N_1$  exécute le routage intra-grappe pour atteindre  $N_2$ .  $N_1$  et  $N_2$  sont dans la même grappe, ils ont donc une même vue locale. Ainsi, ils prendront des décisions de routage cohérentes.

Une telle solution ne crée pas de boucle de routage : le paquet se rapproche à chaque fois d'un saut de la destination. Cependant, il est possible d'avoir des incohérences dans les tables de voisinages, créées de façon distribuée [WU 04]. Pour éviter les boucles, un paquet déjà vu est simplement supprimé silencieusement. Un nœud considère qu'il a déjà traité le paquet si un paquet avec le même couple adresse source/identifiant de paquet a déjà été relayé. Le calcul de la route pour traverser une grappe est fait dynamiquement. Un nœud choisit toujours de relayer le paquet vers la première grappe connue, il doit donc mettre à jour la route du *Route Reply* s'il la modifie, de telle manière que la source bénéficie de ce calcul de route dynamique. Pour un paquet de données, une telle modification n'est pas requise puisque la source ne recevra pas la modification. Finalement, la longueur de nos routes n'est pas optimale, mais nous essayons de la limiter en relayant le paquet à la grappe connue la plus proche de la destination. Un tel calcul dynamique de route est très robuste : les paquets arrivent à destination, même si de nombreux nœuds individuels se déplacent. Il suffit que la route de grappes soit valide pour que le paquet soit acheminé.

### 3.4. Réparation de routes

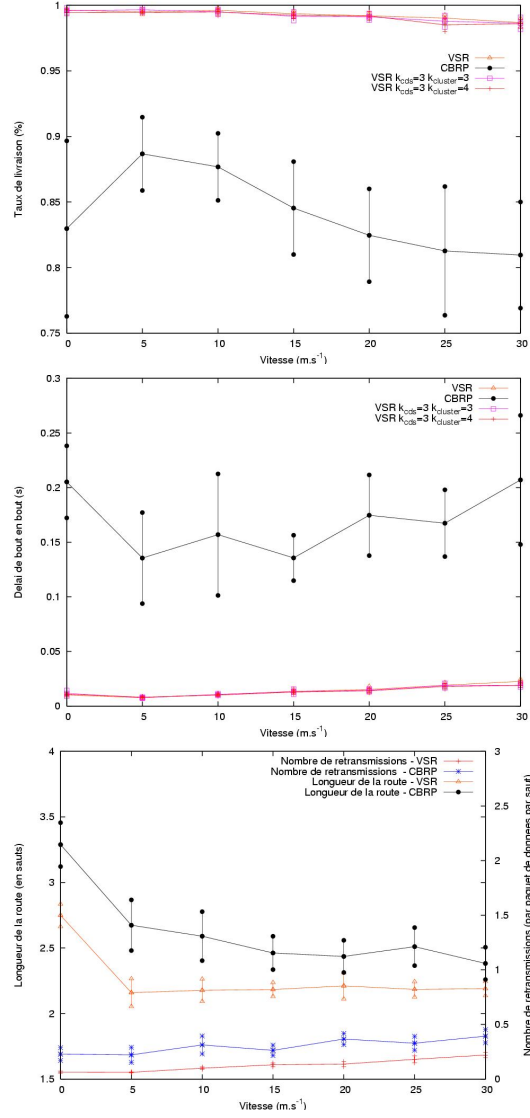
Puisque le taux de livraison est un critère majeur de performance d'un protocole de routage, nous souhaitons le maximiser. Nous proposons donc un mécanisme simple d'acquiescement de paquets. Un nœud ayant envoyé un paquet de données et n'ayant reçu aucun acquiescement au bout d'un certain temps va retransmettre le paquet. Un acquiescement passif en utilisant une coopération avec la couche MAC de IEEE 802.11 est également possible, comme DSR [JOH 03]. Dans ce cas là, aucun overhead n'est induit. Si un paquet n'arrive pas à être transmis sur un lien, nous proposons, comme CBRP, un mécanisme de réparation locale de route. Un tel nœud ré exécute l'algorithme de routage, en interdisant simplement comme prochain saut le nœud fautif. Une telle reconstruction permet de limiter l'impact du délai de convergence des tables de voisinage, améliorant le taux de livraison, mais pénalisant le délai de bout en bout.

## 4. Performances

Nous avons évalué les performances de notre solution en l'implémentant sous Opnet Modeler, avec le modèle 802.11b d'une portée radio de 300m. La trajectoire des nœuds a été simulée avec le modèle de mobilité *random waypoint* sur une surface dépendant du nombre de nœuds pour garder une densité constante. Les paramètres par défaut sont 40 nœuds, un degré de 10, une vitesse de  $5\text{m.s}^{-1}$ . Un flux de données est constitué de 8 paquets de 128 octets, envoyés tous les 0,25 secondes. Le temps inter arrivée des flux suit une loi exponentielle de moyenne 4 secondes. La topologie proposée utilise  $k_{c_{ds}}=2 / k_{cluster}=3$ . Mais, à des fins de comparaison, nous avons simulé également  $k_{c_{ds}}=3 / k_{cluster}=3$  et  $k_{c_{ds}}=3 / k_{cluster}=4$ . Pour des raisons de généralité, nous supposons que les acquiescements doivent être actifs. Comme dans [JIA 99], nous avons fixé  $timeout_{RREQ}$  à 0,5s. Nous avons au maximum une réparation locale de route par nœud.

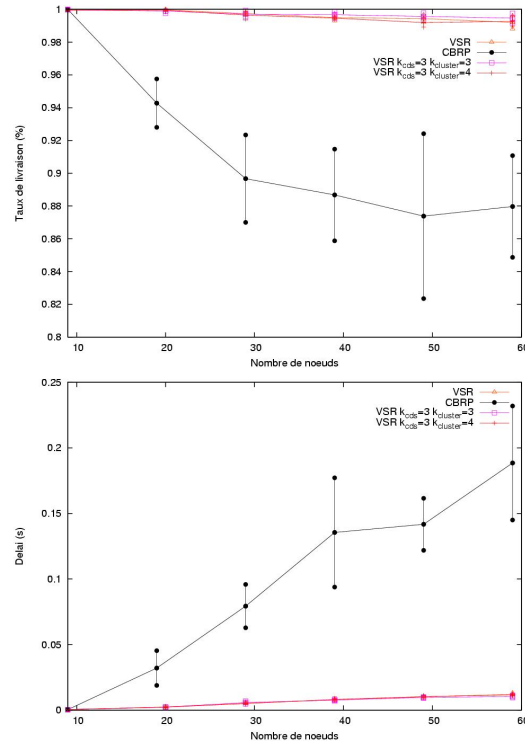
Nous avons évalué les performances de notre solution VSR selon la mobilité, le nombre de nœuds et la charge du réseau. Les principaux paramètres de performances d'un protocole de routage sont selon nous le délai de bout en bout, le taux de livraison, la longueur de la route, (et l'overhead. Nous avons comparé VSR à CBRP puisqu'il est, à notre connaissance, le protocole de routage sur grappes le plus performant.

**Mobilité** Nous avons dans un premier temps évalué la robustesse de notre solution vis à vis de la mobilité (fig. 2). Nous pouvons voir qu'avec VSR le taux de livraison est peu sensible à la mobilité et demeure supérieur à 94%. Les routes sont basées sur une liste de grappe à suivre, elles sont donc robustes. De même le délai reste acceptable, même si une mobilité importante augmente le nombre de reconstructions. Par contre, CBRP voit ses performances fortement impactées par la vitesse. Les routes sont moins robustes puisque basées sur une suite de nœuds-relais individuels à parcourir. Elles cassent donc plus souvent. Le délai de livraison souffre également de ces reconstructions. Les routes sont en moyennes plus longues pour CBRP. En effet,



**Figure 2.** *Impact de la mobilité sur le délai, le taux de livraison et la robustesse des routes*

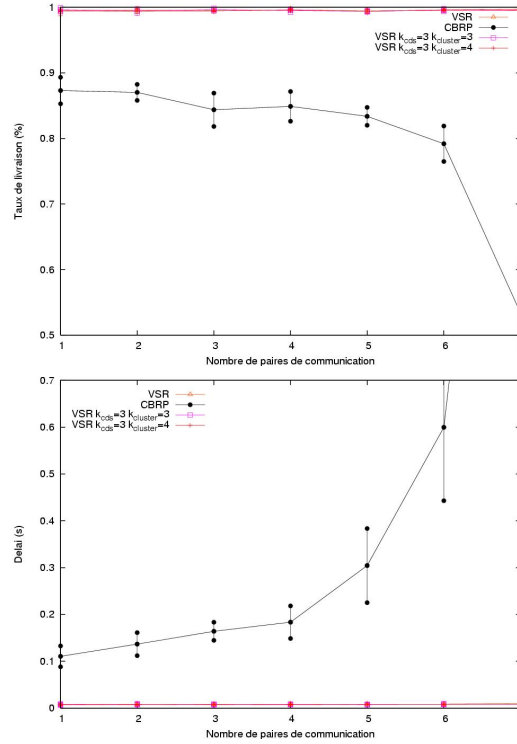
la reconstruction d'une route dans CBRP allonge d'un saut sa longueur, alors qu'elle peut rester optimale dans VSR. De plus, VSR cherche à optimiser la longueur de la route lors d'un Route Reply avec une connaissance locale étendue. CBRP a une connaissance locale limitée, l'optimisation de la longueur semble donc moins efficace.



**Figure 3.** Impact du nombre de participants sur le délai et le taux de livraison

**Passage à l'échelle horizontal** Il est également important qu'un protocole passe à l'échelle horizontalement, i.e. l'augmentation du nombre de nœuds dans le réseau ne doit pas écrouler les performances globales. Nous observons que le taux de livraison de VSR diminue légèrement et le délai augmente, les routes étant en moyenne plus longues (fig. 3). Par contre, CBRP présente des baisses de performances importantes. Les découvertes de routes sont diffusées dans tout le réseau, selon un processus d'inondation très redondant. Le nombre de paquets ainsi générés a donc tendance à créer des pertes dans les paquets de données. Des reconstructions peuvent être tentées, augmentant ainsi le délai, mais ayant néanmoins un impact sur le taux de livraison.

**Passage à l'échelle vertical** Nous avons également évalué l'impact de la charge du réseau, c'est à dire le nombre de communications simultanées que peut écouler le réseau (fig. 4). VSR présente un bon passage à l'échelle. En effet, notre solution crée un overhead résiduel dû au routage local proactif. Cependant, ce coût est mutualisé entre tous les nœuds en communication, l'overhead n'augmentant donc que très peu avec le nombre de découvertes de routes. Par contre CBRP voit ses performances baisser de façon drastique. En effet, le nombre de découvertes de routes générées est



**Figure 4.** Impact de la charge du réseau sur le délai et le taux de livraison

important, la charge réseau créée peut donc ainsi faire perdre des Route Replies, ce qui crée de nouvelles découvertes de routes. Parallèlement, des paquets de données sont perdus, les routes semblent ainsi casser, générant des reconstructions et de nouvelles découvertes de routes. Finalement, une tempête d'inondation se produit, ayant un impact très important sur les performances globales.

**Overhead** Nous avons mesuré l'overhead induit par les 2 solutions. Nous calculons l'overhead comme le nombre de paquets de contrôle par nœuds par seconde. VSR crée un overhead important dû à la construction/maintenance de la structure. Cependant, l'overhead restant est négligeable, les découvertes de routes générant un nombre de paquets de contrôle très réduit par rapport à CBRP. Notre topologie virtuelle permet donc un meilleur passage à l'échelle vertical : une charge plus importante du réseau génère plus de découvertes de routes, mais le nombre de paquets correspondant est très réduit. CBRP a un overhead dominé par la découverte de routes, qui augmente lorsque plus de nœuds participent au réseau, ou que plus de connexions sont initiées. Lorsque la charge est faible (3 communications simultanées), CBRP génère légèrement moins d'overhead que VSR. Par contre, lorsqu'il existe 7 communications si-

multanées, VSR possède un overhead très réduit par rapport à CBRP qui engendre une tempête de broadcast. Par manque de place, nous n'avons pas reporté les résultats en terme d'overhead généré pour des valeurs différentes de rayon de cluster et de dorsale. De manière générale, plus le rayon de cluster est grand, plus l'overhead généré est grand puisque les paquets hellos générés sont relayés plus loin.

|      | Nb Connexions | Hellos | Topologie | RREQ  | RREP  | Acks | Total |
|------|---------------|--------|-----------|-------|-------|------|-------|
| VSR  | 3             | 3,9    | 0,21      | 0,005 | 0,003 | 0,58 | 4,7   |
| CBRP |               | 0,52   | 0         | 1,92  | 0,23  | 1,4  | 4,1   |
| VSR  | 7             | 3,9    | 0,42      | 0,010 | 0,005 | 1,34 | 5,67  |
| CBRP |               | 0,59   | 0         | 6,96  | 0,59  | 2,67 | 10,81 |

## 5. Conclusion

Nous avons proposé un nouveau protocole de routage hybride et hiérarchique pour les réseaux ad hoc et hybrides reposant sur une auto-organisation du réseau. Il utilise une dorsale pour aider notamment à limiter l'overhead induit par une découverte de route, problématique dans les solutions de routage à plat. En effet, seuls les nœuds de la dorsale relaient un paquet de requête de route. La dorsale en forme d'arbre remplit donc parfaitement son rôle de collecteur de trafic de contrôle, point faible des réseaux ad hoc. Parallèlement, nous avons défini de véritables zones de service, possédant chacune un coordinateur de zone. Ces chefs permettent de structurer le réseau en créant une topologie plus stable, et donc des routes plus robustes. Nous avons proposé un protocole de routage intra-grappe proactif, intégré à la découverte de voisinage nécessaire à la maintenance de la topologie virtuelle. Le routage inter-grappe est réactif pour limiter les échanges de paquets de topologie. Notre solution hybride est de plus entièrement paramétrable : si nous choisissons un rayon de grappe important, le proactif sera principalement utilisé. Si au contraire le rayon est plus petit, les découvertes de routes seront plus fréquentes. Nous pouvons penser à une solution adaptant dynamiquement la valeur de  $k_{cluster}$  en fonction de son environnement. Le compromis entre réactif et proactif, entre délais, overhead et taux de livraison est donc dans notre solution complètement adaptable. Par ailleurs, les routes sont calculées dynamiquement, tant pour le routage intra-grappe que pour le routage inter-grappe. Les routes sont donc tirées des informations locales les plus récentes, et présentent donc une plus grande robustesse. Enfin, nous proposons un mécanisme de reconstruction local de routes pour améliorer le taux de livraison.

Notre topologie virtuelle constitue un véritable socle pour implémenter de nouvelles fonctionnalités évoluées telles que le routage, l'assignation d'adresse, le paging, la gestion de la micro-mobilité dans un réseau hybride. . Il serait intéressant de proposer une solution de routage multicast, utilisant la forme d'arbre de la dorsale. Une solution de routage avec qualité de service pourrait également être étudiée. Enfin, notre solution structurant le réseau, créant une hiérarchie tirant partie de l'hétérogénéité naturelle d'un réseau ad hoc, nous pourrions également penser à une solution d'éco-



nomie d'énergie, permettant aux dominés de s'endormir, puisque seuls les dominants possèdent une fonction de coordination essentielle au fonctionnement du réseau.

## 6. Bibliographie

- [CLA 03] CLAUSEN T., JACQUET P., « Optimized Link State Routing Protocol (OLSR) », RFC n° 3626, October 2003, IETF.
- [HAA 98] HAAS Z. J., PEARLMAN M. R., « The Performance of Query Control Schemes for the Zone Routing Protocol », *SIGCOMM'98*, Vancouver, Canada, September 1998, ACM.
- [HED 88] HEDRICK C., « RIP version 2 », RFC n° 2453, June 1988, IETF.
- [JIA 99] JIANG M., LI J., TAY Y. C., « Cluster Based Routing Protocol (CBRP) », Internet draft n° version 01, July 1999, IETF.
- [JOH 03] JOHNSON D. B., MALTZ D. A., HU Y.-C., « The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) », Internet draft n° version 09, April 2003, IETF.
- [MOY 98] MOY J., « OSPF Version 2 », RFC n° 2328, April 1998, IETF.
- [NI 99] NI S., TSENG Y., CHEN Y., SHEU J., « The Broadcast Storm Problem in a Mobile Ad hoc Network », *MobiCom*, Seattle, USA, August 1999, ACM, p. 151-162.
- [PEA 99] PEARLMAN M. R., HAAS Z. J., « Determining the Optimal Configuration of the Zone Routing Protocol », *IEEE Journal on Selected Areas in Communications*, vol. 17, n° 8, 1999, p. 1395–1414.
- [PER 94] PERKINS C. E., « Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers », *SIGCOMM*, London, United Kingdom, August 1994, ACM, p. 234–244.
- [PER 03] PERKINS C. E., BELDING ROYER E. M., DAS S. R., « Ad hoc On-Demand Distance Vector (AODV) Routing », RFC n° 3561, July 2003, IETF.
- [SIN 99] SINHA P., SIVAKUMAR R., BHARGHAVAN V., « CEDAR : a Core-Extraction Distributed Ad Hoc Routing Algorithm », *INFOCOM*, New York, USA, March 1999, IEEE, p. 202–209.
- [SIV 98] SIVAKUMAR R., DAS B., BHARGHAVAN V., « Spine routing in ad hoc networks », *Cluster Computing*, vol. 1, n° 2, 1998, p. 237–248.
- [THE 04a] THEOLEYRE F., VALOIS F., « Robustness and Reliability for Virtual Topologies in Wireless Multihops Access Networks », *Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, June 2004.
- [THE 04b] THEOLEYRE F., VALOIS F., « A Virtual Structure for Mobility Management in Hybrid Networks », *Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, March 2004, IEEE, p. 1035–1040.
- [WU 04] WU J., DAI F., « Mobility Management and Its Applications in Efficient Broadcasting in Mobile Ad Hoc Networks », *Infocom*, Hong Kong, march 2004, IEEE.